A detection method of lost assets based on feature optimization and active-passive detection

Jingchen Yan, Chenxi Cai, Zhe Du, Jianbin Li*
School of Control and Computer Engineering, North China Electric Power University, Beijing, China

ABSTRACT

With the development of Internet technology, various network attacks have emerged one after another, seriously affecting the security of many key infrastructures such as finance, energy, and transportation. Therefore, the importance of network asset management is self-evident. How to judge the security of assets and detect lost assets has become an important research topic. This paper proposes a method for detecting lost assets based on feature optimization and active-passive detection. Firstly, it achieves the classification of abnormal traffic by extracting important features of the traffic data. And then, it detects the network assets using the combined active and passive detection method. The experiments show that this method can effectively detect the lost assets in the network and effectively provide an analysis basis for threat analysis and emergency response.

Keywords: Lost assets, feature optimization, active-passive detection

1. INTRODUCTION

From ARPAnet[1] in the United States in the 1960s to today's Internet, network technology has developed rapidly, and more and more organizations and individuals are accessing the Internet. With the continuous expansion of network scale, network assets have greatly promoted production and office efficiency, but they have also brought many security problems and security risks[2]. Various network attacks have emerged one after another, seriously affecting the security of many key infrastructures such as finance, energy, and transportation and causing huge economic losses[3]. A timely grasp of the network assets security is an essential prerequisite for network security and lays the foundation for threat correlation analysis. Therefore, detecting lost assets has become an important research topic.

Traditional asset detection is mainly aimed at physical network equipment, such as Web servers, routing equipment, etc[4]. The definition of asset in ISO/IEC 13335-1:2004 is "anything of value to the organization"[5], that is, in addition to physical equipment, information systems, network services, other responsible units, and related personnel information are also important assets that cannot be ignored[6]. This paper mainly discusses network assets such as terminals, equipment, and services that have network connections among the above assets.

Although there are many researches on network asset detection[7-8], most of these methods have lag and imperfections, and it is challenging to obtain real-time information efficiently and accurately about dynamically changing network asset status[9], timely discover lost assets, and minimize the loss caused by security issues.

Therefore, this paper proposes a method for detecting lost assets based on feature optimization and active-passive detection. Firstly, extract the essential features of abnormal traffic and classify the abnormal traffic accurately based on the features. Secondly, combine active and passive detection methods to detect the corresponding lost assets. This method can quickly and accurately locate the lost assets, respond to terminal threats, and reduce the damage caused by the attack. It provides an essential technical foundation for the security defense of network attacks.

The paper structure is as follows: Section 2 introduces related research work and summarizes the shortcomings of existing methods. The third section presents the method of detecting lost assets based on feature optimization and active-passive detection in detail. Section 4 presents the comparative experiments and the results. Finally, we summarize this work.

*lijb87@ncepu.edu.cn

International Conference on Computer Application and Information Security (ICCAIS 2021), edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12260, 1226008 · © The Authors. Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2637519

2. RELATED WORK

Asset detection refers to the effective identification, extraction, and conversion of network asset information involved by analyzing the traffic data during operation and combining it with the drawn correlation graph to complete attack behavior analysis and security threat prevention. At present, the technology of detecting network assets has been developed to a certain extent. The traditional methods of detecting network assets are mainly manual statistics and semi-automatic statistics relying on the client[10]. Conventional identification methods are not applicable for a system network with multiple nodes and huge assets. The current mainstream methods include active scanning detection[11], passive learning detection [6], non-invasive detection of search engines[12], and so on. AI-Shehari et al.[13] proposed a system asset detection method based on the C4.5 decision tree algorithm classifier. This method uses the hash value of the TCP connection socket to associate the SYN packet with the FIN packet and compare the SYN packet in the p0f fingerprint library. The device fingerprint is expanded to improve the accuracy of network asset detection and traceability. Tyagi et al.[14] proposed an operating system equipment asset detection method based on Euclidean distance, which is used to discover unauthorized abnormal equipment hosts in the system, which shortens the modeling time compared to other complex classifiers. Reference[15] used the C4.5 decision tree model to achieve passive detection of device fingerprints based on TCP/IP protocol stack with a shorter modeling time and higher accuracy, which improves the detection rate of fingerprints that are not accurately matched. This method improves the accuracy of network asset identification and traceability. Simon et al.[16] comprehensively used the target domain name information obtained by Google and Shodan to realize the domain name's non-invasive asset detection and vulnerability analysis. Cyberspace has the characteristics of multi-layered, dynamic, and highly complex. In the face of the constantly updated security situation, effectively perceiving the situation changes in cyberspace in real-time is very important, especially for detecting lost assets.

Abnormal traffic detection refers to finding abnormal traffic patterns that are inconsistent with the expected behavior from the data. The lost assets usually implement the abnormal traffic pattern in the system. The detection and analysis of the abnormal traffic in the system can improve the identification ability of the lost assets. Attack has the characteristics of concealment, antagonism, and complexity. The collected information also contains many false and misoperation data. Because of the above problems, the anomaly detection for traffic data has some problems, such as content missing, feature redundancy, and low accuracy. Existing research technologies include machine learning such as Bayesian networks, decision trees, support vector machines, deep learning such as autoencoders, deep belief networks, convolutional neural networks, and cyclic neural networks. Erfani et al.[17] used Belief Network (DBN) to extract common underlying features, used a single-class SVM to learn features from DBN, and finally completed anomaly detection. Sarvari et al.[18] proposed an abnormal traffic detection method based on abnormal points, using an improved cuckoo search algorithm, a mutant cuckoo fuzzy algorithm for abnormal traffic feature selection, and an evolutionary neural network for behavior classification and recognition. Reference[19] proposed an abnormal traffic detection method based on hidden Markov statistical analysis. This method first collects the traffic data's IP characteristics, inputs the hidden Markov model for modeling, and then uses the conditional entropy parameter optimization algorithm to speed up the hidden Markov model parameter update and improve the parameter optimization rate. Reference[20] proposed an abnormal traffic detection method based on stacked noise reduction autoencoders. The method first uses the particle swarm algorithm to optimize the number of hidden layers and nodes of the SDAE structure and then uses the gradient descent method to train and learn the optimized SDAE. According to the detection accuracy of the training samples, the optimal flow characteristics are extracted. Finally, the flow feature is classified by adding a SoftMax classifier to detect abnormal flow. The current abnormal traffic detection work mainly results from feature selection optimization and algorithm capability improvement. Although good performance has been achieved, there is still room for further optimization. Accurately filtering out abnormal traffic from massive data to perform high-precision analysis of lost assets is a current research difficulty.

In general, the existing methods for detecting lost assets still have many problems, such as the imbalance of data samples of fallen assets, the redundancy of characteristic traffic dimensions, and the backwardness of model methods. These problems lead to low accuracy and long-time delay in detecting lost assets. Therefore, it is imminent to carry out accurate and efficient research on detecting lost assets.

3. THE DETECTION METHOD OF LOST ASSETS BASED ON FEATURE OPTIMIZATION AND ACTIVE-PASSIVE DETECTION

3.1 Overview

This paper constructs a lost asset detection model based on feature optimization and active-passive detection, as shown in Figure 1. The model includes an abnormal traffic data detection module and a lost asset detection module. Firstly, process the traffic data to obtain fixed-length traffic data with rich features but low dimensionality, and then get important abnormal traffic features through feature importance analysis. After that, use BiGRU[21] model to classify abnormal traffic. At last, obtain intranet traffic through network mirror port technology, and use a combination of active and passive detection methods to detect lost assets.

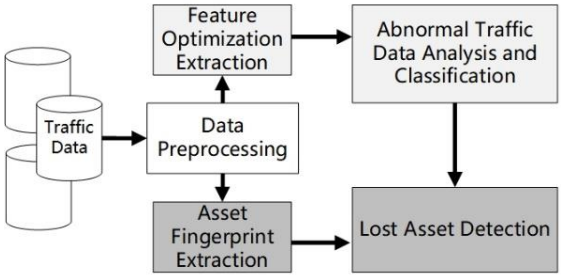

Figure 1. The architecture of the lost asset detection model.

3.2 Abnormal traffic data detection module

Abnormal traffic data detection mainly distinguishes the abnormal situation and category of the current system through the analysis and extraction of the input characteristics of the traffic data. The existing methods are primarily suitable for processing fixed-length and low-dimensional traffic data[22]. The artificial selection of feature selection is highly subjective, leading to high time delay, high energy consumption, and low accuracy. Therefore, this paper proposes an abnormal traffic data detection model, and the process is shown in Figure 2. The model is based on the spatial pyramid pooling theory[23], which adds a feature processing layer to the feature learning process. It can reconstruct the features of the original input data so that the model can process variable-length input and redundant input. After that, the features' importance in the new data set is analyzed through the self-attention mechanism. The features are weighted according to the analysis results to filter out the important features of abnormal traffic data. At last, the traffic data detection based on BiGRU is used to classify the traffic with the above-mentioned important characteristics as normal and abnormal.

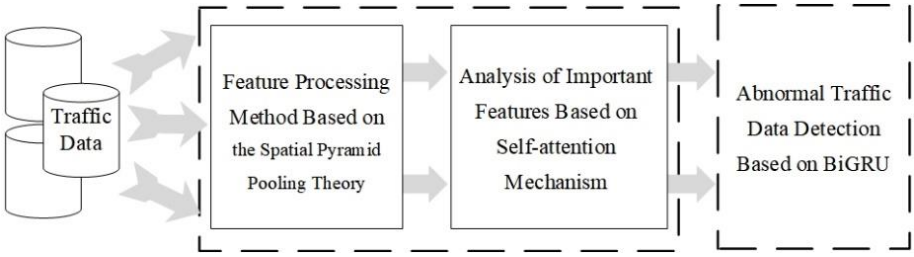

Figure 2. The architecture of the abnormal traffic data detection module.

- Normalize the collected traffic data. The null value is filled based on the context information, and the traffic data is appropriately scaled to reduce the dimensional influence between the feature columns, as shown in equation (1). The characteristic column ϑ, the i-th element in ϑ_i, $\vartheta_{\max}$ and $\vartheta_{\min}$ are the maximum and minimum values of the column, that is

$$\vartheta_i = \frac{\vartheta_i - \vartheta_{\min}}{\vartheta_{\max} - \vartheta_{\min}} \tag{1}$$

- Select convolutional neural networks as the infrastructure to construct the feature learning module according to the spatial feature of traffic data. The feature processing layer is added between the last convolution layer and the full

connection layer to replace the original maximum pool layer to realize the dimension reconstruction of variable length input data features. Set multiple pooled cores, and set the size of a single pooled core to n/f dimension according to the dimension n of input data and the dimension f of output data. The feature processing layer has *a* pool cores. The pooled core output is spliced to form the final output dimension, as shown in equation (2).

$$\mathrm{F}_{data} = \sum_{j=1}^{a} f(\mathrm{F}_{indata}^{Conke\_layer}, \mathrm{C}_j) \tag{2}$$

$\mathrm{F}_{indata}^{Conke\_layer}$ is the preliminarily processed data obtained by the initial input data after the convolution operation of the convolutional layer. The function represents inputting the data F into j pooling cores for processing and aggregation. By introducing a feature processing layer and performing data feature extraction at different levels, it is possible to achieve data reduction and reconstruction while retaining the original data features to the maximum extent, reducing computational complexity, and improving model training capabilities. After the above operations, we have obtained fixed-length traffic data with rich attribute features but low dimensionality.

- To improve the feature learning ability of the abnormal traffic data detection model, the self-attention mechanism analyses and evaluates the feature importance of the traffic data. The model learns the input data information, scores, weighs each feature dimension of the input and highlights the impact of important features on downstream training modules. As shown in equation (3), we map the data F_{data} to three different spaces to obtain the query vector Q_F, key vector K_F, and value vector V_F, respectively, and then use matrix operations to perform dot product operations to obtain the feature importance score matrix $Feature_{F_{data}}^{weight}$. Since the gradient may become very small when the value is too large during the calculation process, we add d_k to prevent the gradient from disappearing, where d_k is equal to the number of hidden layers divided by the number of self-attention heads. Finally, we weighted the feature importance score matrix $Feature_{F_{data}}^{weight}$ with the data set F_{data} to obtain a new data set F_{weight_data}, as shown in equation (4).

$$Feature_{F_{data}}^{weight} = soft\max(\mathrm{Q}_F * \mathrm{K}_F^T / \sqrt{d_k}) * \mathrm{V}_F \tag{3}$$

$$F_{weight\_\mathrm{data}} = Feature_{F_{data}}^{weight} * F_{data} \tag{4}$$

- Analyze and detect abnormal traffic using BiGRU on the new data set. The obtained abnormal traffic data will be transmitted to the next module for in-depth analysis of the lost assets.

3.3 Lost asset detection module

This section introduces the method of lost assets detection through the active-passive detection methods. Combine Nmap and Masscan tools to detect the operating system information and service application or component information of network lost assets. In terms of passive detection, this paper proposes an improved random forest method to classify and detect the operating system information and service applications of lost assets and build a database of lost assets information, as shown in Figure 3:

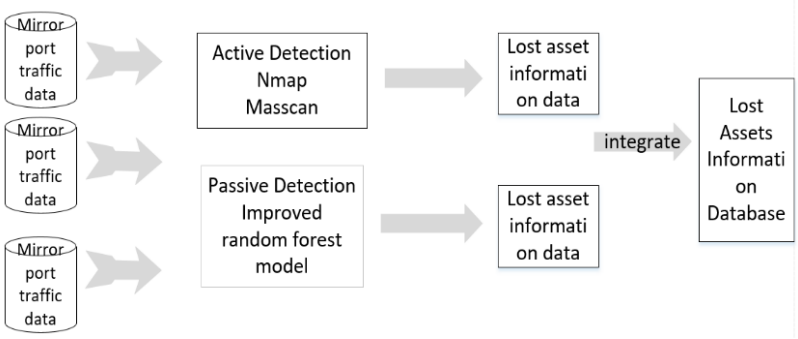

Figure 3. Lost asset detection module.

The steps of the active detection method are as follows:

- Firstly, use mirror port technology to collect data streams of abnormal behaviors, and record their IP address information and abnormal behaviours.
- Call the python-nmap module and python-masscan module in the python library, and use the IP address information to perform host discovery and port scanning for devices that may be lost assets. If the device is alive and has many open ports, use nmap for services and operating system detection.
- Build a database of lost assets information from the processed results.

The steps of the passive detection method are as follows:

- Since the CURE-SMOTE algorithm[24] can reduce the noise introduced to address the problem of an imbalanced data sample, we use the Random Forest model before the divided subsets, using the CURE-SMOTE sampling algorithm to obtain a new data set.
- Gaussian mixture model decision tree clustering algorithm on the precision stage is generated clusters, using Gaussian mixture model EM algorithm iterates, thus ensuring parameters within an iterative process always converge to a local optimum.

$$p(x) = \sum_{k=1}^{k} \pi_k N(x \mid \mu_k, \sum k) \tag{5}$$

To reduce the time overhead of the random forest model, we adopt a parallelization method to reduce the time to build a decision tree. The specific steps of parallelization are as follows:

- Select training samples for each decision tree from the updated total data set;
- For each machine, construct an attribute sub-list for the specified number of Map functions as a subset of candidate split attributes processed by each Map function;
- Each Map function calculates the information gain and Gini coefficient of all possible split points of the corresponding attribute sub-list and returns the key-value pair <key, list<values>>;
- At the same time, for the node, call the Reduce function to count a series of key-value pairs returned in step (3), and select the optimal split value as the split attribute and classification point of the node;
- For each node of the tree, repeat steps (2) to (4), and stop if the attribute space is empty or the training set is empty;
- All constructs complete decision tree classifier is written on the Hadoop Distributed File System to get the final Random Forest classifier.

4. EXPERIMENT AND RESULTS

The following experiments are carried out to verify the effectiveness of the method proposed in this paper.

4.1 Experimental environment and evaluation indicators

The hardware environment used in this experiment: Ubuntu 16.04 LTS system, Intel Core i7 8750H CPU, 16G memory. Software environment: Python 3.6, JDK1.8, Hadoop-2.6.0, Keras, Scikit-learn. Mirror port technology is adopted, and network packet software "Wireshark" collects network traffic generated by networked devices.

To evaluate the model constructed in this paper from many aspects, this paper adopts $\mathrm{Precision}$, $Recall$ and $F_1 - score$ performance evaluation of the model. The $\mathrm{Precision}$ rate, $Recall$ rate and $F_1 - score$ are defined as follows:

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Among them, TP represents the number of types a sample that is correctly predicted to type a, FP is the number of non-types a sample that are incorrectly predicted to type a, and FN is the number of type a sample that are incorrectly predicted to be non-a type. In the two-class experiment, category a refers to the abnormal category, F_1 takes the weighted average.

4.2 Dataset

Commonly used public datasets for abnormal traffic detection include CTU-13, CIC IDS2017, UNSW_NB15, and so on. The dataset used in this article is UNSW_NB15, which was created in 2015 by the IXIA PerfectStorm tool in the Australian Centre for Cyber Security (ACCS) laboratory. Regular traffic and nine types of attack traffic appeared in the dataset: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The original pcap package is provided in the dataset, and a CSV file with tags and extracted features is also provided.

The lost asset dataset comes from the complete network asset data formed by the asset information data collected by the active detection and passive detection of the network assets in our laboratory.

4.3 Model realization

The overall architecture of the abnormal traffic detection module and the missing asset detection module has been explained in detail in Section 3. This unit describes the specific implementation of the model.

In the process of network model realization, the methods provided by functions such as Conv1D(), MaxPooling1D(), and Dense() provided by the Keras package are used to construct a 1D-CNN model. To avoid overfitting, a dropout layer is added. The Dropout layer will randomly choose to ignore hidden layer nodes so that the network trained each time is different, and finally use the same weight for fusion. The Relu function is selected for the activation function is that the function does not have a saturation zone and does not have the problem of gradient disappearance, does not involve complex exponential calculations, and has high computational efficiency and fast convergence. Some main parameter settings are shown in Table 1:

Table 1. Abnormal traffic detection module experimental parameter information.

Parameter	Description	Settings
num_filters	Convolutional layer filter size	48
kernel_size	Convolution kernel size	3
pooling_size	Pooling layer size	2
Dense	Fully connected layer size	16
activation	Activation function	Relu
hidden_size	BiGRU hidden layer number	128
Learning_rate	Learning rate	0.01

In the active detection phase, use the python-nmap and python-masscan modules of the Nmap and Masscan tools in the python library to implement ping scanning, port scanning, service, and version detection. Some key pseudo-codes are as follows:

Algorithm 1 Use Nmap for asset detection

1: **for** each i ∈ nm.all_hosts() **do**

2: **Output** host:hostname()

3: **Output** state:state()

4: **for** each a ∈ all_protocols() **do**

5: **Output** Protocol: proto ()

6: **Make** Lport value equal to nm[host][proto]. keys () value

7: Lport sort ()

8: **for** each n ∈ lport **do**

9: **Output** all port information and services

The pseudo-codes of the decision tree establishment and the training process of the random forest algorithm used in the passive detection phase are as follows:

Algorithm 2 Decision tree establishment

Input: TrainData D={(x_1, y_1), (x_2, y_2), …, (x_m, y_m)};

Attribute Set A={a_1, a_2, …, a_d};

Procedure: Function TreeGenerate(D, A) ;

1: **Generate nodes** node;

2: **if** All samples in D belong to the same category C **then**

3: **Make** node as a C-type leaf node; **return**

4: **end if**

5: **if** A= ∅ OR The samples in D have the same value on A **then**

6: **Make** node as a leaf node, and mark its category as the category with the largest number of samples in D; **return**

7: **end if**

8: Select the optimal partition attribute a_* from A ;

9: **for** each value of a_* a_*^v **do**

10: **Generate** a branch for node; **let** D_v denote the subset of samples in D whose value is a_*^v on a_*;

11: **if** D_v=Null **then**

12: The branch node is marked as a leaf node, and its category is marked as the category with the most samples in D; return

13: **else**

14: **Make** TreeGenerate(D_v, A\ {a_*}) as the branch node

15: **end if**

16: **end for**

Output: A decision tree with the node as the root node

4.4 Experimental results

There are two types of abnormal traffic detection results: normal traffic and abnormal traffic, two classification problems. In the research of abnormal traffic detection, commonly used analysis methods include decision trees, integrated algorithms, neural networks, etc. We selected several typical methods to compare with the method in this paper. Mainly from the precision rate (Precision), recall rate (Recall), and F1-score analysis, the results are shown in Table 2. When detecting abnormal traffic, the method proposed in this paper has an accuracy rate of 0.968 and a recall rate of 0.952, which is the best among many methods.

Table 2. Comparison of abnormal traffic detection methods.

Logistic REGRESSION	Adaboost	CNN	CNN-BiLSTM	Our method
0.874	0.891	0.936	0.953	0.968
0.855	0.880	0.911	0.946	0.952
0.864	0.885	0.923	0.949	0.960

From the first piece of data in Table 3, the recall rate of the sample set of type OS reached 1, and the accuracy rate was 90%, indicating that all the samples were correctly predicted. It is difficult to achieve a perfect balance between the accuracy and recall rates of various types of network assets. Therefore, appropriate performance metrics should be selected for the importance of recall and precision in actual situations.

Table 3. Detection results of our method.

Network asset	Precision	Recall	F1-score
OS	0.90	1.00	0.95
Server	0.82	0.88	0.85
Printer	0.45	0.55	0.50
Switch	0.86	0.98	0.92
Router	0.73	0.62	0.67
Gateway	0.78	0.53	0.63
ED	0.85	0.70	0.77

In summary, experiments show that the method proposed in this paper can effectively detect lost assets.

5. CONCLUSIONS

This paper proposes a method for detecting lost assets based on feature optimization and active-passive detection. This method can accurately classify abnormal traffic through traffic data analysis and detect the lost assets corresponding to the anomalous traffic data, which provides a basis for threat analysis and emergency response.

We did not evaluate UNSW_Nb15 training focused on balancing normal and malicious samples in the experiment. In the future, we will try to use oversampling, undersampling, and other technologies to expand the samples with fewer samples or reduce the number of samples, to obtain better test results.

REFERENCES

[1] Hauben, M., “History of arpanet,” Site de l’Instituto Superior de Engenharia do Porto, 17, 1-20 (2007).

[2] Kim, A., Oh, J., Ryu, J. and Lee, K., “A review of insider threat detection approaches with IoT perspective,” IEEE Access, 8, 78847-78867 (2020).

[3] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P. and Sikdar, B., "A survey on IoT security: Application areas, security threats, and solution architectures," IEEE Access, 7, 82721-82743 (2019).
[4] Reddy, R. A., Swamy, J, and Reddy, R. G., "Detecting embedded devices using network discovery," International Journal of Innovative Science and Modern Engineering (IJISME), 2(5), 27-32 (2014).
[5] Murphy, C. N. and Yates, J., [The International Organization for Standardization (ISO): Global Governance through Voluntary Consensus], Taylor & Francis, New York, 5-24 (2009).
[6] Sanders, C. and Smith, J., [Applied Network Security Monitoring: Collection, Detection, and Analysis], Elsevier, 1-496 (2013).
[7] Harris, D., Hyde, R. and Smith, D., "Network and asset management: Benefits of real-time data," NZ Transport Agency Research Report No. 638, 10-83 (2018).
[8] Chen, S., "Network security protection technology under the background of computing big data," Journal of Physics: Conference Series, 1982(1), 012207 (2021).
[9] Genge, B., Graur, F. and Enăchescu, C., "Non-intrusive techniques for vulnerability assessment of services in distributed systems," Procedia Technology, 19, 12-19 (2015).
[10] Scheel, P. C., [IDS-Based Passive Asset Detection], University of Oslo, Oslo, Master's Thesis, (2014).
[11] Lyon, G. F., [Nmap Network Scanning: The official Nmap Project Guide to Network Discovery and Security Scanning], Insecure. Com LLC (US), California, 150-465 (2008).
[12] Chen, Y., Lian, X., Yu, D., Lv, S., Hao, S. and Ma, Y., "Exploring shodan from the perspective of industrial control systems," IEEE Access, 8, 75359-75369 (2020).
[13] AlShehari, T. and Shahzad, F., "Improving operating system fingerprinting using machine learning techniques," International Journal of Computer Theory and Engineering, 6(1), 57 (2014).
[14] Tyagi, R., Paul, T., Manoj, B. and Thanudas, B., "Packet inspection for unauthorized of detection in enterprises," IEEE Security & Privacy, 13(4), 60-65 (2015).
[15] Yi, Y. H., Liu, H. F. and Zhu, Z. X., "Research of passive OS recognition based on decision tree," Computer Science, 43(8), 79-83 (2016). (in Chinese)
[16] Simon, K., Moucha, C. and Keller, J., "Contactless vulnerability analysis using google and shodan," J. Univers. Comput. Sci., 23(4), 404-430 (2017).
[17] Erfani, S. M., Rajasegarar, S., Karunasekera, S. and Leckie, C., "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," Pattern Recognition, 58, 121-134 (2016).
[18] Sarvari, S., Sani, N. F. M., Hanapi, Z. M. and Abdullah, M. T., "An efficient anomaly intrusion detection method with feature selection and evolutionary neural network," IEEE Access, 8, 70651-70663 (2020).
[19] Xiao, L. and Wang, H., "Network intrusion detection based on hidden Markov model and conditional entropy," Inter. Conf. on Smart City and Informatization, 509-519 (2019).
[20] Dong, S. and Zhang, B., "Network traffic anomaly detection method based on deep features learning Journal of Electronics and Information," 42(3), 695-703 (2020).
[21] Yin, X., Liu, C. and Fang, X., "Sentiment analysis based on BIGRU information enhancement," Journal of Physics: Conference Series, 1748(3), 032054 (2021).
[22] Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J., "Survey of intrusion detection systems: Techniques, datasets and challenges," Cyber Security, 2(1), 1-22 (2019).
[23] He, K., Zhang, X., Ren, S. and Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904-1916 (2015).
[24] Ma, L., [Research on Optimization and Improvement of Random Forests Algorithm], Jinan University, Jinan, Master Thesis, (2016). (in Chinese)